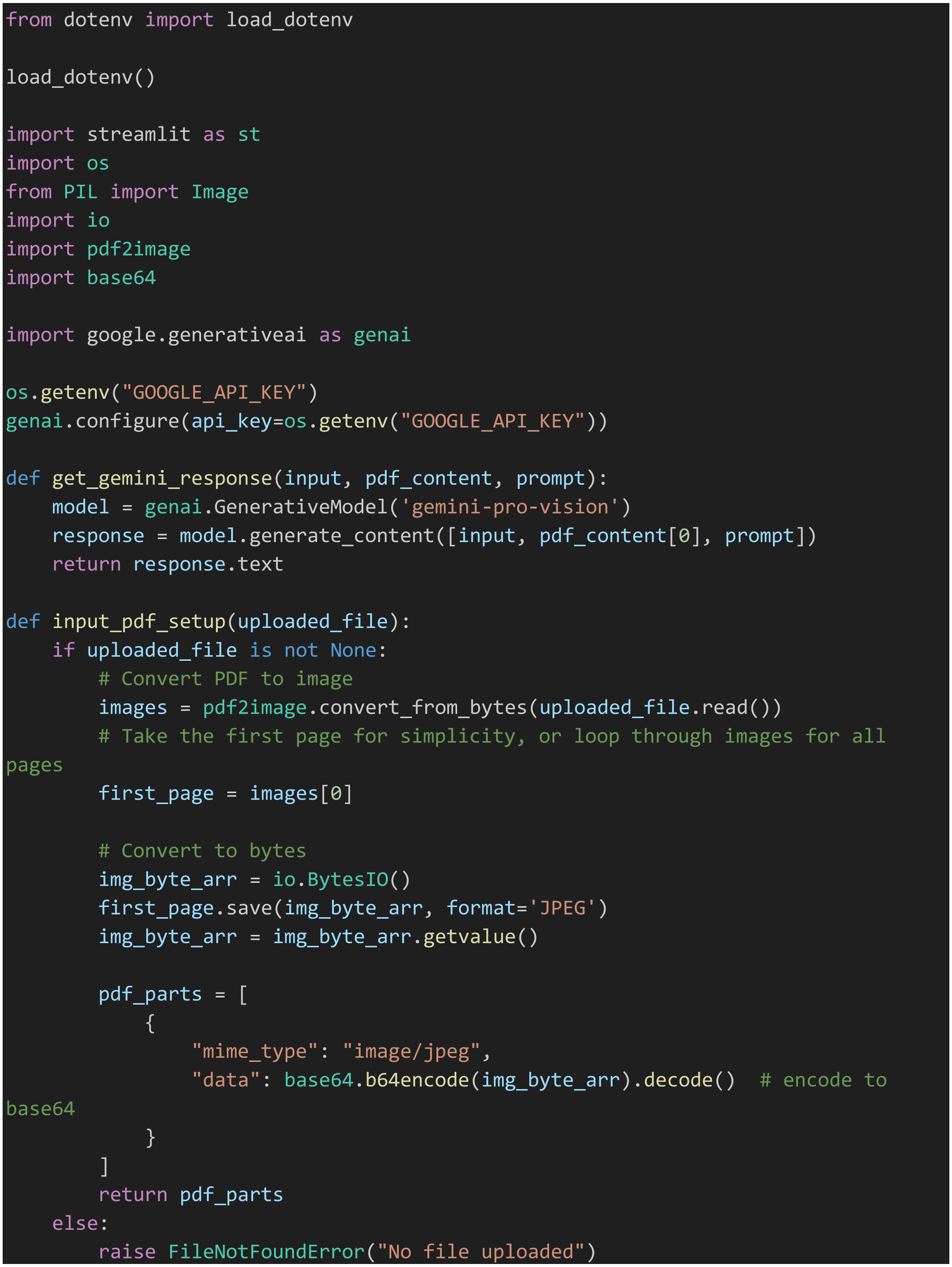
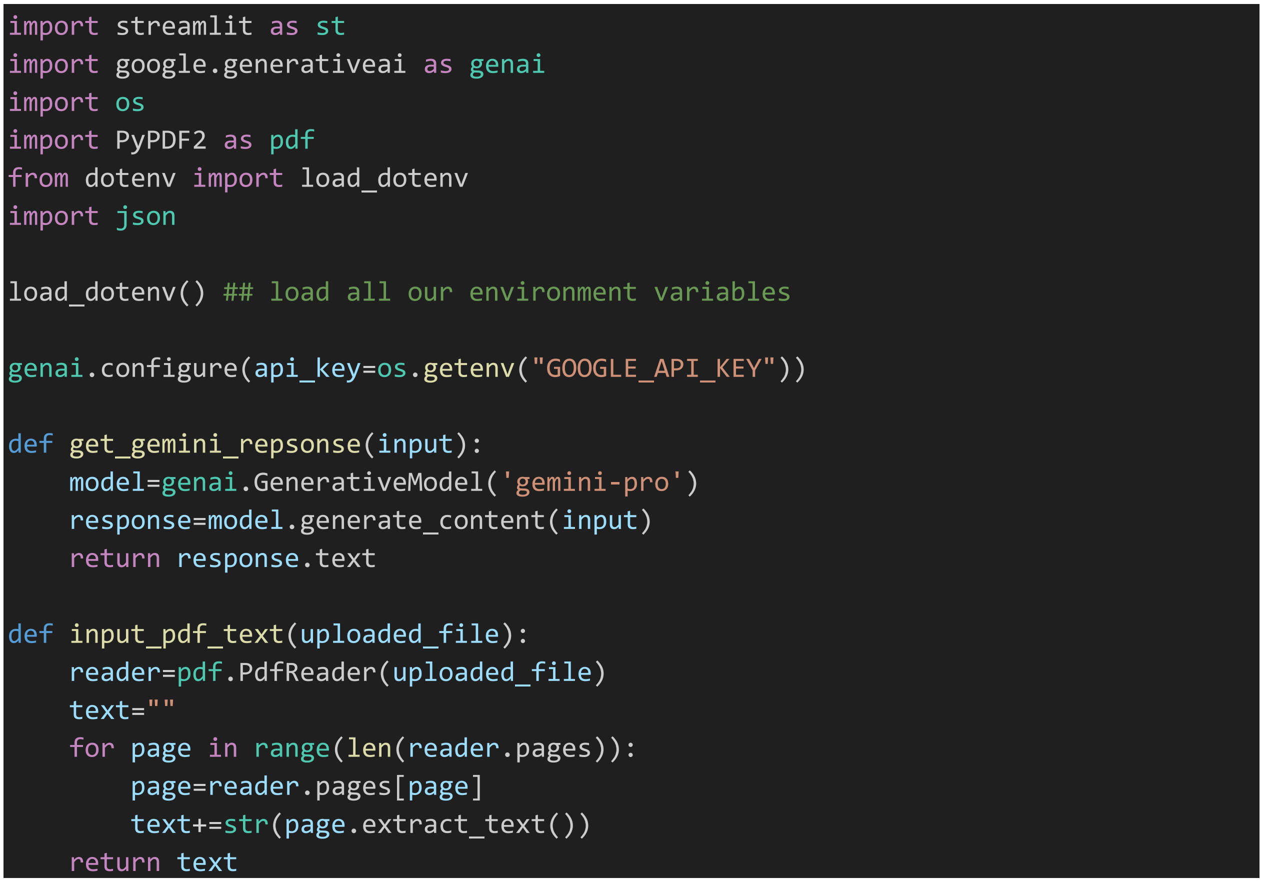


```
from dotenv import load_dotenv

load_dotenv()

import streamlit as st
import os
from PIL import Image
import io
import pdf2image
import base64

import google.generativeai as genai

os.getenv("GOOGLE_API_KEY")
genai.configure(api_key=os.getenv("GOOGLE_API_KEY"))

def get_gemini_response(input, pdf_content, prompt):
    model = genai.GenerativeModel('gemini-pro-vision')
    response = model.generate_content([input, pdf_content[0], prompt])
    return response.text

def input_pdf_setup(uploaded_file):
    if uploaded_file is not None:
        # Convert PDF to image
        images = pdf2image.convert_from_bytes(uploaded_file.read())
        # Take the first page for simplicity, or loop through images for all
pages
        first_page = images[0]

        # Convert to bytes
        img_byte_arr = io.BytesIO()
        first_page.save(img_byte_arr, format='JPEG')
        img_byte_arr = img_byte_arr.getvalue()

        pdf_parts = [
            {
                "mime_type": "image/jpeg",
                "data": base64.b64encode(img_byte_arr).decode()  # encode to
base64
            }
        ]
        return pdf_parts
    else:
        raise FileNotFoundError("No file uploaded")
```



```
import streamlit as st
import google.generativeai as genai
import os
import PyPDF2 as pdf
from dotenv import load_dotenv
import json

load_dotenv() ## load all our environment variables

genai.configure(api_key=os.getenv("GOOGLE_API_KEY"))

def get_gemini_repsonse(input):
    model=genai.GenerativeModel('gemini-pro')
    response=model.generate_content(input)
    return response.text

def input_pdf_text(uploaded_file):
    reader=pdf.PdfReader(uploaded_file)
    text=""
    for page in range(len(reader.pages)):
        page=reader.pages[page]
        text+=str(page.extract_text())
    return text
```